

Intro To Python Exercises

Remember to save all of the programs you will be working on using the file extension `.py`. Backup ALL Python programming files on your Google Drive by the end of every class. Make a folder where you can save all of your programs. All you need to do is drag your files into your browser tab when you have Google Drive opened. At home, it may be easier to use Google File Stream to help organize your files:

<https://support.google.com/a/answer/7491144?hl=en>

Before getting started, let's look at how to comment your code. In Scratch, you had to right click on a block of code to add comments. Here, you have two options: use `#` for a single line comment or triple quotes for commenting using multiple lines.

Single Line Comments

The `#` sign is for comments. A comment is a line of text that Python won't try to run as code. It's just for humans to read. Comments make your program easier to understand. When you look back at your code or others want to collaborate with you, they can read your comments and easily figure out what your code does. You must always comment lines of code that introduce variables. You need to specify their purpose in the code. You also need to comment parts of code as well. Every line of code does not need to be commented. You can write comments for chunks of code to explain what a section is specifically doing.

Multi-Line Comments

For multi-line comments, you can include the whole block in a set of triple quotation marks. Example:

```
"""  
  
Here is where you can write multiple sentences  
to explain what your code is doing.  
  
"""
```

Topic: Math and Documentation

Header

Every one of your programs going forward must have what is called a header. Create a sample header for your own programs, and save the file as "Header.py". You will be using this header in all programs going forward. There is an example below but make sure to make yours is generic enough so you can modify it for each new program.

Intro To Python Exercises

"""

Filename: printName.py

Author: Jane Doe

Date: 2017-11-29

Description: prints user's name to the screen

"""

All programs below will need to be submitted. Submit all files at the very end to the Google Classroom. Every file must have a detailed header. Remember to update your generic copy of a header each time you write a new program. You will have multiple .py files to hand in so be sure to save a backup periodically.

Program #1: Type in this program line by line, exactly as it appears. Remember to put your header in first at the very top of your program. This is an example of how to use variables and output text.

```
#variables demonstrated
print ('This program is a demo of variables')
v = 1
print ('The value of v is now ', v )
v = v + 1
print ('v now equals itself plus one, making it worth', v )
v = 51
print ('v can store any numerical value, to be used elsewhere.')
print ('for example, in a sentence. v is now worth', v )
print ('v times 5 equals', v*5)
print ('but v still only remains', v)
print ('to make v five times bigger, you would have to type v = v * 5')
v = v * 5
print ('there you go, now v equals', v, 'and not', v / 5)
```

Add in this extra line where you will be using string concatenation (this refers to the joining together of strings):

```
print ("This is the end of the " + "program." + "\n" + "Goodbye")
```

The “\n” is a newline character. It formats the output so that “Goodbye” is printed on a new line. Python also does addition (+), subtraction (-), division (/), exponential calculations (e.g. 2**3 meaning 2 to the power of 3 which equals 8) and modulus (%). We covered modulus in Scratch when determining if a number was even or odd.

Save your file as YourFirstName_001.py (Example: If your first name is Joe, your filename should be Joe_001.py)

Upload the new file to your Google Drive folder as well. Do this periodically to avoid losing your work. You can attach your work to the Google Classroom at this point but do not click on Turn In until all programs are completed.

Intro To Python Exercises

Program #2: Create a program that calculates the sum (addition) of 5, 6, and 7.4, including text explaining what you are doing. An example has been pasted below with different numbers. Make sure you use 5, 6, and 7.4. Save the file as **YourFirstName_002.py**

```
#the str() forces the information inside the parentheses to become a string
print ('The sum of 1,2, and 3 is: ' + str(1+2+3))
```

Program #3: Create a program that calculates the quotient (division) of 5 and 2 and store the result in a variable with a specific name. Use a single line comment and make a prediction as to what you think the output will be. Then run the program and see what the real output is. Is it the same? Why or why not? Answer this question in your comment. Then print the result in a well formatted sentence that clearly explains what calculation was done and what the final answer is. Use the variable name in the print statement. Save the file as **YourFirstName_003.py**

Program #4: You can copy your code from program #3 but this time, you will be dividing 5 by 2.0. Store the result in a variable with a specific name. Use a single line comment and make a prediction as to what you think the output will be. Then run the program and see what the real output is. Is it the same? Why or why not? Answer this question in your comment. Then print the result in a well formatted sentence that clearly explains what calculation was done and what the final answer is. Use the variable name in the print statement. Save the file as **YourFirstName_004.py**

Program #5: Create a program that uses BEDMAS in order to calculate an answer. Your program should print out the equation, and the calculated answer to the equation. Your equations to work with is here: $3 \times ((2 + 12) - 2 \times (5 - 1))$. Here is a sample of what the code could look like with a different equation:

```
print ('1 + 2 x 3 / (4 + 2) = ' + str(1+2*3/(4+2)))
```

Remember to use this equation: $3 \times ((2 + 12) - 2 \times (5 - 1))$

Save your file as **YourFirstName_005.py**

Note: Keyboard Input

Review the example below to help with the next program. We are using Python version 3, so `raw_input` will not work. Use `input` instead of `raw_input`.

```
#Getting a string value from the user
name = raw_input("Enter your name: ")

#Getting an integer value from the user
age = int(raw_input("Enter your age: ")) #the int( ) forces the value to be an integer

#Getting a float value from the user
salary = float(raw_input("Enter your salary: "))

#Outputting information based on the values above
print ('Hello, ' + name + '. I noticed that you are ' + str(age) +
      ' years old, and have a salary of ' + str(salary) + '.')
```

Intro To Python Exercises

Program #6: Create a program that asks the user for 3 integer inputs, then outputs the sum of the numbers. Remember to use variables and then print a message to the screen that says what the three numbers were followed by the result. Save your file as **YourFirstName_006.py**

Program #7: Create a program that prompts for the length, width, and height of a box. Calculate and output the surface area and the volume of this box. Remember to print out specific sentences that lists the dimensions and what the surface area and volume are. Save the file as **YourFirstName_007.py**

Program #8: Create a program that prompts for a name, address, city, country, and postal code. Use appropriate variable names for all variables. Print out the information as though it were an address label. See http://rlv.zcache.ca/canada_maple_leaf_flag_gray_and_black_design_label-ra2922e7f29604aa5af6d84b0298c203b_v1130_8byvr_324.jpg as a reference. Save the file as **YourFirstName_008.py**

Now you are completely finished. Upload all eight of your files to the Google Classroom and then click on “Turn In”.

Remember to save a copy of your files on your Google Drive as well.